SMM638 Final Course Project Description

Simone Santoni

2024-11-25

Synopsis

This notebook illustrates the final course project for the SMM638 course. The project is based on the analysis of a dataset regarding the friendship network and genre preferences of Deezer users. The first section of the notebook provides an overview of the project, the second section describes the data, the third section outlines the problem, and the fourth section describes the submission package.

# Overview

Like other streaming platforms, Deezer contains a wealth of digital traces, which can be used to analyze user behavior, and, therefore, to create or refine products and improve business model execution (e.g., by adopting a recommendation system that help a platform business better engage with audiences).

Network analysis methods and tools play a key role when it comes to analyzing digital-traces like the one we have in the Deezer dataset. Particularly, network analysis offers an effective framework within which to appreciate the similarity between entities — being users or the genres they may favorite — and, possibly, cluster these entities into homogenous groups — e.g., users that share similar music genres or genres that are liked by the same users. Let us consider a two-mode (or bipartite) network , where users are connected to genres via the ‘like’ relationship:

where is the ‘like’ relationship between user and genre . The matrix can be used to create a user-user network ( x ) and a genre-genre network ( x ):

The user-user network is a one-mode, non-directed, weighted graph where nodes are users and edges are mutual likes, i.e., the counts of music genres that users and share. The genre-genre network is a one-mode, non-directed, weighted graph where nodes are genres and edges are mutual likers, i.e., the counts of users that like both genres and . Consider the following example of ‘like’ network, including five users and three music genres:

The user-user network () is:

whereas the genre-genre network () is:

Both and can be further analyzed using network analysis tools — e.g., block-modeling — or conventional statistical tools — e.g., cluster analysis – to identify homogenous groups of entities (users and genres for and , respectively).

# Data

The data for the final course project is stored in the [data/deezer\_clean\_data](https://github.com/simoneSantoni/net-analysis-smm638/tree/master/data/deezer_clean_data) directory of [GitHub repository of SMM638](https://github.com/simoneSantoni/net-analysis-smm638). The data, which were gathered for a network science project,[[1]](#footnote-23) are also available in the website of [Stanford Network Analysis Project](https://snap.stanford.edu/data/gemsec-Deezer.html).

Below are some key aspects about the data:

* The data were scraped from Deezer in November 2017
* \*\*\_edges.csv represent friendships networks of users from 3 European countries, that is, Croatia, Hungary, and Romania. Nodes represent the users and edges are the mutual friendships[[2]](#footnote-25)
* \*\*\_genres.json contain the genre preferences of users — each key is a user identifier, the genres loved are given as lists. Genre notations are consistent across users. In each dataset users could like 84 distinct genres. Liked genre lists were compiled based on the liked song lists

## Friendship networks

For illustrative purposes, let us inspect the friendship network for the case of Croatia. First, we load Pandas and NetworkX, then we load the data:

# load modules  
import pandas as pd  
import networkx as nx  
# load data  
fr = pd.read\_csv('../data/deezer\_clean\_data/HR\_edges.csv')  
# data preview  
fr.head()

|  | node\_1 | node\_2 |
| --- | --- | --- |
| 0 | 0 | 4076 |
| 1 | 0 | 29861 |
| 2 | 0 | 53717 |
| 3 | 0 | 23820 |
| 4 | 0 | 39945 |

The data preview shows that the friendship network for Croatia is a list of edges, where each edge is a pair of user identifiers. The data can be used to create a network object using NetworkX:

fr\_g = nx.from\_pandas\_edgelist(fr, source='node\_1', target='node\_2')  
fr\_g?

Using code introspection, it is possible to see that the network object fr\_g is a NetworkX object of type Graph and that it has 54,573 nodes and 498,202 edges. To familiarize with the data, we test if fr\_g is connected:

nx.is\_connected(fr\_g)

True

Then, we consider the degree distribution of the network:

# import further modules  
import numpy as np  
from matplotlib import pyplot as plt  
from collections import Counter  
# compute node degree  
dd = Counter(dict(fr\_g.degree()).values())  
# plot the degree distribution  
fig = plt.figure(figsize=(4, 3))  
ax = fig.add\_subplot(111)  
ax.scatter(dd.keys(), dd.values(), color="limegreen", alpha=0.15)  
ax.set\_yscale("log")  
ax.set\_xscale("log")  
ax.set\_xlabel("Log(Degree)")  
ax.set\_ylabel("Log(Counts of nodes)")  
ax.grid(True, ls="--")  
plt.show()

|  |
| --- |
| Figure 1: Degree distribution of the friendship network for Croatia |

It is self-explanatory that the degree distribution of the friendship network for Croatia is right-skewed, which is a common feature of social networks. We can try to getter a better understanding of the network — including the presence and locatio of ‘hub’ users — by visualizing it. Since the network is large, we may benefit from using the visualization capabilities of [graph-tool](https://graph-tool.skewed.de/), a Python API wrapping around C++ code, a more efficient alternative to pure Python NetworkX:

# import further module  
from graph\_tool.all import \*  
# iterate over the Pandas DataFrame to create the graph and edges to it  
edges = [(str(u), str(v)) for u, v in fr[['node\_1', 'node\_2']].values]  
fer\_gt = Graph(edges, hashed=True, directed=False)  
# plot the network  
# graph\_tool.draw.graph\_draw(fer\_gt, output\_size=(500, 500), output="fer\_gt.png")  
# load image  
from IPython.display import Image  
Image(filename='fer\_gt.png')

|  |
| --- |
| Figure 2: Friendship network for Croatia (N=54,573) |

It is worth noticing the friendship network presents a periphery of users with low degree and, plausibly, a core of users with high degree. However, the figure does not provide a clear picture of the core of the network, which deserves further investigation.

## Music genre preferences

Building on the previous sub-section, we consider the preferences of users as per HR\_genres.json files. These files are JSON files, which can be loaded using the json module:

import json  
with open('../data/deezer\_clean\_data/HR\_genres.json', 'r') as f:  
 pr\_json = json.load(f)  
pr\_json["11542"]

['Indie Rock',  
 'Indie Pop/Folk',  
 'International Pop',  
 'Rap/Hip Hop',  
 'Pop',  
 'Rock',  
 'Indie Pop',  
 'Alternative']

At this stage, we have a dictionary where each key is a user identifier and the corresponding value is a list of genres that the user likes. For example, above is the list of music genres that user 11542 likes. We can convert the dictionary into a Pandas DataFrame drawing upon Pandas’ json\_normalize function:

pr = pd.json\_normalize(pr\_json).T  
pr.rename({0: 'genres'}, axis=1, inplace=True)  
pr.head()

|  | genres |
| --- | --- |
| 13357 | [Pop] |
| 11542 | [Indie Rock, Indie Pop/Folk, International Pop... |
| 11543 | [Dance, Pop, Rock] |
| 11540 | [International Pop, Jazz, Pop] |
| 11541 | [Rap/Hip Hop] |

The data preview shows that the DataFrame pr has a single column, genres, which contains lists of genres that users like. To make the data more amenable to analysis, we can explode the lists of genres into separate rows drawing upon Pandas’ explode function:

pr = pr.explode('genres')  
pr.reset\_index(inplace=True)  
pr.rename({'index': 'user\_id'}, axis=1, inplace=True)  
pr.head()

|  | user\_id | genres |
| --- | --- | --- |
| 0 | 13357 | Pop |
| 1 | 11542 | Indie Rock |
| 2 | 11542 | Indie Pop/Folk |
| 3 | 11542 | International Pop |
| 4 | 11542 | Rap/Hip Hop |

For illustrative purposes, we can consider the distribution of genres liked by users in the dataset:

genres = Counter(pr.groupby('genres').size())  
fig = plt.figure(figsize=(6, 3))  
ax = fig.add\_subplot(111)  
ax.hist(genres.keys(), color="magenta", alpha=0.5)  
ax.set\_xticklabels(["{:,}".format(int(x)) for x in ax.get\_xticks()])  
ax.set\_xlabel("Degree -- number of likers")  
ax.set\_ylabel("Counts of music genres")  
ax.grid(True, ls="--")  
plt.show()

|  |
| --- |
| Figure 3: Degree distribution for music genres in the ‘Croatia’ dataset |

The degree distribution of music genres liked by users is right-skewed, meaning there are few genres that are liked by many users and many genres that are liked by few users.

# Problem description

Your employer is a consultancy firm that has been hired by ‘Sonic,’ a major music label, to better understand the categories that compose the music market. The [artists and repertoire](https://en.wikipedia.org/wiki/Artists_and_repertoire) roles at Sonic have struggled to make sense of the association between the ‘music genre’ tags (e.g., ‘International Pop’) in Deezer and similar services. Some think that these tags are sometimes redundant. In other circumstances, they are unclear or meaningless. Therefore, it is hard for Sonic to see clear targets in the market, and, consequently, correctly position new albums and musicians against consumer preferences. Sonic wants a map of the categories that form the music market. This map must be based on some digital traces – i.e., behavioral data – , easy to interpret, and well-grounded in demonstrable data patterns.

To help Sonic address its business problem, you have been asked to analyze the ‘Croatia’ dataset, briefly described in the previous section. You are expected to use network analytic methods and tools to:

1. Assess the similarity between Deezer music genres
2. Identify homogenous groups of Deezer music genres
3. Highlight how social ties among users influence the similarity between music genres

# Submission package

The submission package consists of:

* A report that includes:
  + The description of the workflow you have followed to address the problem (300 words MAX)
  + The results of your analysis, comprising text (300 words MAX) and exhibits (five among figures and tables MAX)
  + The interpretation of the results in light of the business problem (300 words MAX)
* The computer code that allows me to fully reproduce your charts (being, R, Python, Julia, Rust, C++, Java, etc.). The code should be well-commented and easy to read. Non-reproducible exhibits will not be graded.

1. Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. 2018. GEMSEC: Graph Embedding with Self Clustering. arXiv preprint arXiv:1802.03997. [↑](#footnote-ref-23)
2. The researchers who collected the data say that they have “*… reindexed the nodes in order to achieve a certain level of anonymity*.” [↑](#footnote-ref-25)